

L U M B R E



Unir y transformar tablas

Grado: 2

Materia: Programación con Python para Datos · Programación

11/10/2026

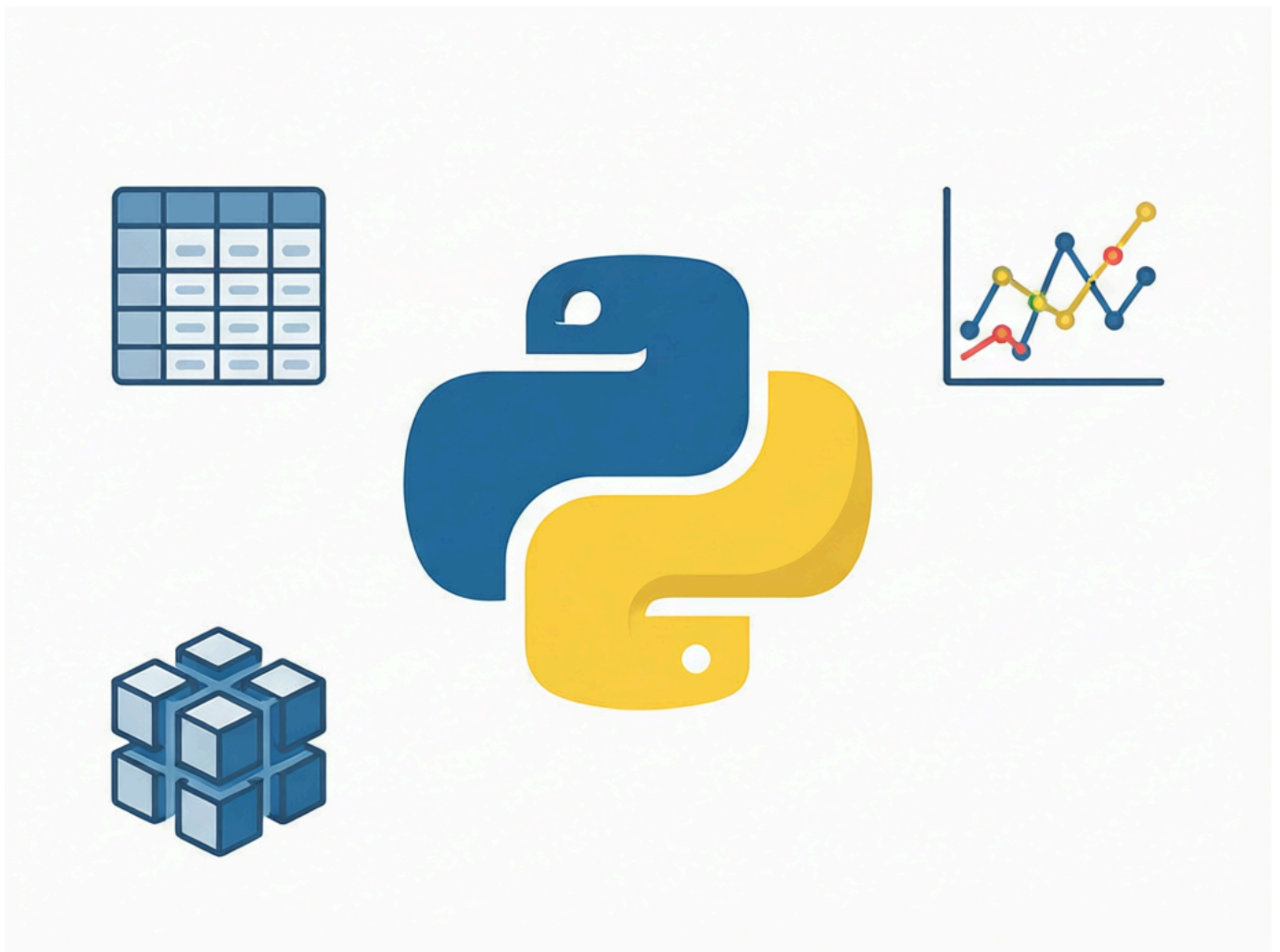
Antes de empezar

Los datos rara vez vienen en una sola tabla: hay que unirlos y a veces reorientarlos de ancho a largo. Dominar merge, concat, melt y pivot es poder montar el análisis a partir de piezas.

QUÉ APRENDERÁS

- Combinar DataFrames con concat y merge, y reorganizar su forma con melt y pivot.

Unir y transformar tablas



Python para ciencia de datos: NumPy, pandas y manipulación de datos.

Mapa conceptual

- **Python para Datos**
 - Entorno
 - pip

- venv
- Reproducibilidad
- Funcional
 - Comprensiones
 - Lambda
 - map filter
- Calidad
 - try except
 - pytest
 - Asserts
- NumPy
 - Arrays
 - Broadcasting
 - Indexacion
- Pandas
 - DataFrame
 - Groupby
 - Merge Join

El mapa mental interactivo es interactivo en la versión online.



Ensamblar y reorientar

Una vez tienes datos limpios por separado, el siguiente paso es ensamblarlos: unir tablas que comparten claves y reorientar las que vienen en la forma equivocada. Aquí es donde el análisis cobra sentido.

concat: apilar

`pd.concat` apila DataFrames. Con `axis=0` añade filas (varios meses del mismo informe); con `axis=1` añade columnas lado a lado. Es la operación de «juntar trozos del mismo tipo».

merge: relacionar

`pd.merge` une dos tablas por el valor de una o más columnas clave, como un JOIN de base de datos. Cada fila de una se empareja con las de la otra que comparten clave,

enriqueciendo la información.

Tipos de join

- inner: solo las claves presentes en ambas tablas.
- left: conserva todas las filas de la izquierda, con NaN donde la otra no tenga pareja.
- right y outer: simétricos y completo, respectivamente.

Elegir el join correcto

La pregunta es qué NO quieres perder. Si necesitas cada pedido aunque no puedas completar el cliente, usa left sobre pedidos. Si solo te interesa donde hay datos en ambos lados, inner. El tipo de join decide el número de filas del resultado y, con él, la conclusión.

Cada necesidad de ensamblaje tiene su operación.

Situación	Herramienta	Efecto
Añadir filas del mismo esquema	concat(axis=0)	más largo
Añadir columnas alineadas	concat(axis=1)	más ancho
Unir por una clave común	merge(on=...)	ancho enriquecido
Conservar aunque falte pareja	merge(how=left)	huecos como NaN

1. Necesitas una tabla con TODOS los clientes y sus pedidos, incluyendo los clientes que no han pedido nada. ¿Qué merge usas?

- ☐ inner join entre clientes y pedidos
- ☐ left join con clientes a la izquierda
- ☐ concat de las dos tablas
- ☐ cross join

Claves con nombres distintos

A veces la columna clave se llama diferente en cada tabla: merge(left_on=id_cliente, right_on=cliente_id) lo resuelve. Tras unir, suelen sobrar columnas duplicadas que conviene descartar.

Peligro de las duplicidades

Si la clave se repite en ambos lados, merge produce el producto de las coincidencias y el resultado crece más de lo esperado. Validar que la clave es única (o entender su granularidad) antes de unir evita tablas infladas.

melt: de ancho a largo

Un DataFrame «ancho» (una columna por mes) es difícil de agrupar. `pd.melt` lo vuelve «largo»: cada medida pasa a una fila con una columna de variable y otra de valor. Es la forma que prefieren `groupby` y las librerías de gráficos.

pivot: de largo a ancho

`pivot` (y `pivot_table`) hace lo inverso: convierte valores de una columna en cabeceras, cruzando con índices y agregando si hay repetidos. Convierte un listado largo en una matriz legible por celdas.

Cuándo reorientar

- Datos mensuales en columnas que quieres analizar como serie: melt a largo.
- Un listado de transacciones que quieres ver como tabla por categoría y mes: `pivot_table`.
- Preparar un informe: `pivot`; preparar para agrupar o graficar: `melt`.

Errores frecuentes

- Unir sin comprobar la unicidad de la clave y duplicar filas sin darse cuenta.
- Elegir `inner` cuando se necesitaban las filas no coincidentes, perdiendo datos en silencio.
- Confundir `concat` (apilar) con `merge` (relacionar por clave).

2. `pd.merge` puede generar más filas que las tablas de origen si la clave de unión está repetida en ambos lados.

☐ Verdadero

☐ Falso

Cadena de ensamblaje

Un análisis real suele ser: cargar varias fuentes, limpiar, unir por claves, reorientar y luego agregar. Pensar el pipeline en esas etapas hace que cada operación sea pequeña y verificable.

3. Asigna cada operación a su propósito:

Elementos:

- pd.concat
- pd.merge
- pd.pivot
- pd.melt

Apilar	. . .
Relacionar	. . .
AAcho	. . .
ALargo	. . .

Pandas Docs https://pandas.pydata.org/docs/user_guide/merging.html Guía de combinación y reorganización de DataFrames.

VIDEO

Video complementario

Recurso audiovisual para reforzar los conceptos.



Escanea para verlo en Lumbre

<https://www.youtube.com/watch?v=rpscVS0vtbw>

4. Tienes una tabla ancha de ventas por trimestre y una tabla de regiones con su continente. Describe los pasos: ¿reorientarías primero o unirías primero? Justifica el orden según lo que quieras responder.

Documentación pandas <https://pandas.pydata.org/docs/> Guía de referencia para manipulación de datos en Python.

Referencia curricular

Grado:	2
Materia:	Programación con Python para Datos · Programación

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://univ.lumbre.lat/object/cdia-pyth-009-unir-y-transformar-tablas>

Material educativo generado desde Lumbre · lumbre.lat